REMARKS

Claims 1-24 and 27-33 are currently pending in the subject application and are presently under consideration. Claims 1-3, 13, and 28-30 have been amended as shown on pages 2-7 of the Reply.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 1-24, 27-29, and 33 Under 35 U.S.C. §102(e)

Claims 1-24, 27-29, and 33 are rejected under 35 U.S.C. §102(e) as being anticipated by Lewallen (US Pat No. 7,020,882 B1). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Lewallen does not disclose each and every feature set forth in the subject claims.

> For a prior art reference to anticipate, 35 U.S.C. §102 requires that "each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *In re Robertson*, 169 F.3d 743, 745, 49 USPQ2d 1949, 1950 (Fed. Cir. 1999) (*quoting Verdegaal Bros., Inc. v. Union Oil Co.*, 814 F.2d 628, 631, 2 USPQ2d 1051, 1053 (Fed. Cir. 1987)).

The subject application relates generally to remote visualization of an industrial device using Scalable Vector Graphics (SVG). According to one or more embodiments, remote access to a device can be provided *via* a graphical depiction of various device features and related information based on a representation of the device embedded within an associated SVG file. The system can include a Web-based interface that can retrieve one or more Scalable Vector Graphics (SVG) XML markup language files generated for the device from a data store associated with the device. The retrieved SVG file can be executed locally in connection with the Web-based interface or an invoked open software package. The use of SVG files to render an interactive graphical interface can mitigate the need to download image files and Java applets, thereby reducing overhead and bandwidth requirements (see, *e.g.*, page 4, lines 1-15). In particular. In particular, amended independent claim 1 recites, *an interface component that* ***retrieves a Scalable Vector Graphics (SVG) file from storage associated with a device****, the SVG file including data representative of the device's physical faceplate; and a display component*

that ***executes the SVG file within a Web browser via American Standard Code for Information Interchange (ASCII) drawing commands to render an interactive graphical representation of the device's faceplate*** *within a remote viewing window of the Web browser,* ***the interactive graphical representation allowing remote monitoring and modification of at least one parameter associated with the device via the Web browser***.

Contrary to assertions made in the Office Action, Lewallen does not disclose at least these aspects. Lewallen relates to a technique for remote manipulation of a user interface across a network. According to this technique, a server transmits an object, such as Document Object Model (DOM), to a remote computer. A layout engine at the remote computer then generates an initial user interface from the downloaded DOM such as an initial HTML page or other DOM user interface. The server additionally transmits W3C APIs to the remote computer. A bridge at the remote computer then translates the W3C APIs to one or more user interface (UI) APIs that implement the W3C APIs within the user interface. The layout engine then executes the user interface APIs translated from the W3C APIs to manipulate the user interface DOM and generate commands to alter the displayed user interface (see column 11, lines 11-35). The Office Action ostensibly equates the W3C API received at the remote computer with the SVG file of independent claim 1, indicating in particular that the W3C API is converted to a user interface API at the remote computer, and that this user interface can comprise a Scalable Vector Graphic format of the DOM.

However, there are numerous functional distinctions between the system described in Lewallen and that of the subject claims. For one, Lewallen does not disclose that an *SVG file* can be retrieved and executed to render an interactive graphical representation of a device's faceplate. Rather, according to the cited reference, a remote computer downloads a *user interface DOM* and one or more *W3C APIs,* and the W3C APIs are translated to one or more user interface APIs. Although Lewallen states that these resulting user interface APIs can comprise an implementation of the DOM in SVG format, as noted by the Examiner, this SVG-formatted user interface is not the result of *downloading and executing an SVG file*. Rather, the *W3C API* is received at the remote computer and is transformed by the API mappings in the bridge into the appropriately formatted user interface API, which can be in SVG if the remote computer is using such an interface. See, for example, column 11, lines 21-35 of Lewallen, which states:

> "[T]he remote computer receives W3C APIs transmitted over the network from the server. The bridge then *translates the W3C APIs to one or more user interface (UI) APIs* that implement the W3C APIs within the user interface [at the remote computer]. For instance, *the user interface (UI) APIs may comprise the user interface implementation of the DOM, such as the particular implementation of the DOM in Microsoft Internet Explorer 4.0, Netscape Communicator 6.0 and Navigator, Mozilla, the Scalable Vector Graphics format* or any other user interface that implements the DOM specification." (emphasis added)

It is clear from the above passage that no SVG file is retrieved by the remote computer in order to render the interface. Rather, as described in the above passage, the SVG formatted UI API relied upon by the Examiner is the result of transforming a *received W3C API* into an SVG formatted API at the remote computer using the bridge. This transformation is further described at column 5, lines 21-39, and elsewhere in Lewallen. In contrast to the cited reference, amended independent claim 1 provides for *retrieving an SVG file* from storage associated with a device, and *executing the SVG file within a Web browser* to render an interactive graphical representation of the device's faceplate. Retrieving an *SVG file*, rather than downloading and transforming a W3C API as disclosed in Lewallen, can mitigate communication bandwidth usage and reduce computational overhead involved in handling the above-described W3C API transformations. Moreover, direct download of an SVG file to render an interactive graphical representation obviates the need to maintain complicated object mapping definitions as required by Lewallen to effect the API transformations.

Moreover, Lewallen does not disclose that the UI APIs described in that reference *allow remote monitoring and modification of at least one parameter associated with the device* being represented by the graphical representation. Rather, Lewallen describes the user interface at the remote computer generically, and does not identify a particular function being performed by the interface being used at the remote computer. Instead, Lewallen is primarily concerned with a method for remotely manipulating or altering *the user interface itself via* the W3C APIs from the server (see at least column 11, lines 31-39). The Office Action appears to equate this remote modification of the user interface with the remote monitoring and modification of a device parameter as set forth in amended independent claim 1. However, in this regard, the Office Action is relying upon Lewallen's user interface at the remote computer to allege anticipation of the interactive graphical representation of claim 1, while relying upon remote manipulation *performed upon this user interface itself* to read on the aspect of monitoring and modification of

a parameter *by* the interactive graphical representation of claim 1. This inconsistency in interpretation strongly suggests that Lewallen fails to anticipate the rendering and control aspects of amended independent claim 1, especially since, as noted above, Lewallen does not specify a particular function performed by the user interface described in that reference.

Similarly, amended independent claim 13 recites, *a network browser that* ***retrieves an SVG file from the device and executes the file*** *using American Standard Code for information Interchange (ASCII) drawing commands* ***to generate an interactive graphical depiction of the device****, the interactive graphical depiction* ***allowing monitoring and modification of at least one operational parameter within the device.*** Lewallen fails to disclose these features, as already discussed.

Likewise, independent claim 24 recites, *creating a* ***Scalable Vector Graphics (SVG) file*** *that represents at least one aspect of the device; storing the SVG file with the device; employing a remote Web browser to* ***access the SVG file****; and employing American Standard Code for Information Interchange (ASCII) drawing commands to* ***execute instructions embedded within the SVG file at the Web browser to generate an interactive graphical representation*** *of the at least one aspect of the device within the remote web browser,* ***the interactive graphical representation facilitating remote monitoring and modification of at least one operational parameter of the device****.* As discussed *supra*, Lewallen is silent regarding these aspects.

Also, according to one or more embodiments of the present application, the storage employed to store the SVG files for subsequent retrieval can be configured to periodically check for newly created or updated SVG files, and if such a file is located, the SVG file can be automatically retrieved and stored (see, *e.g.*, page 8, lines 26-29). In this way, a client wishing to monitor a remote device can be provided with up-to-date SVG information in order to render a current graphical interface for the device. In particular, claim 4 recites, *the storage associated with the device periodically checks for updated SVG information and automatically retrieves the updated SVG information for storage upon detection.* Lewallen makes no allowances for such an automated updating technique, particularly in the context of the SVG-based remote monitoring system set forth in the subject claims. With regard to these features, the Office Action indicates a passage in Lewallen that merely discusses multithreading (*i.e.* concurrent execution) of multiple mixed statement programs within a browser. The Office Action offers no rationale as to how a general mention of concurrent program execution anticipates *periodically*

checking for updated SVG information, or automatic retrieval of such updated SVG information upon detection, and it is submitted that Lewallen nowhere discloses such functionality.

Likewise, claim 15 recites, *the device-related data is stored in a data bank associated with the device, the data bank periodically checking for updated device-related data and automatically retrieving the updated device-related data for storage upon detection.* Lewallen does not disclose or suggest these features, as already noted.

In view of at least the foregoing, it is respectfully submitted that Lewallen does not disclose each and every feature of amended independent claims 1, 13, and 24 (and all claims depending there from), and as such fails to anticipate or render obvious the present application. It is therefore requested that this rejection be withdrawn.

II. Rejection of Claims 28 and 30-32 Under 35 U.S.C. §102(e)

Claims 28 and 30-32 are rejected under 35 U.S.C. §102(e) as being anticipated by Chapman, *et al.* (US 2004/0021679 A1). It is respectfully submitted that this rejection should be withdrawn for at least the following reasons. Chapman, *et al.* does not disclose each and every feature of the subject claims.

Amended independent claim 28 recites, ***retrieving an SVG file*** *from a computer-readable storage medium associated with the device; and* ***executing the SVG file within the remote interface*** *using American Standard Code for Information Interchange (ASCII) drawing commands to* ***draw a dynamically updated interactive graphic of the device****, the interactive graphic* ***displaying a real-time status of at least one parameter associated with the device and allowing remote modification the at least one parameter***. As discussed in the previous section of the Reply, Lewallen does not disclose or suggest such a method for rendering an interactive graphic of a device. Chapman, *et al.* is also silent regarding these features. Chapman, *et al.* relates to an HMI architecture that facilitates creation of HTML display pages used to remotely visualize an industrial process *via* a Web-based interface. According to Chapman, *et al.*, the display pages act as containers for the page elements that make up the display page and provides the primary user interface thread of execution. These page elements include any one or more elements that can be included *in an HTML file* (see paragraph [0030]). With regard to receipt and execution of an SVG file, the Office Action notes that these HTML files can include Scalable Vector Graphics (see paragraph [0220]). However, employing an *HTML file* that

includes SVG information is not the same as downloading and executing an *SVG file* to render an interactive graphic. The use of an SVG file rather than an HTML file (with or without SVG information) to render an interactive graphic represents an advantage of the present claims over the HTML-based display pages of Chapman, *et al.*, since an SVG file typically requires less bandwidth to download and fewer resources to render than does an HTML file. Chapman, *et al.* does not contemplate retrieval and execution of an SVG file to render an interactive graphic, but rather only discloses embodiments in which *HTML files* are used to generate display pages (see at least paragraph [0201], which states that "use is made of a file format *in as standard an HTML representation as possible*, thus reaping the most benefits from the alignment with industry standards."). As such, the cited reference nowhere discloses or suggests receiving an *SVG file* from a storage medium associated with a device, and executing this retrieved information at a remote interface using ASCII drawing commands to render an interactive graphic of the device.

Similarly, independent claim 32 recites, *means for* ***retrieving a Scalable Vector Graphics (SVG) file with device-related information from a computer-readable storage medium associated with the device****; means for invoking the SVG file within a Web-based browser; means for executing* ***the SVG file within the Web-based browser using ASCII drawing commands to generate an interactive graphical representation of a faceplate for the device****; and means for viewing and modifying at least one operational parameter within the device via the interactive graphical representation.* Chapman, *et al.* does not disclose or suggest generating an interactive representation of a device's faceplate in this manner, as discussed above.

In view of at least the foregoing, it is respectfully submitted that Chapman, *et al.*, alone or in combination with Lewallen, does not disclose each and every aspect of amended independent claims 28 and 32 (and all claims depending there from), and as such fails to anticipate or render obvious the subject application. It is therefore requested that this rejection be withdrawn.

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [ALBRP331USA].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicants' undersigned representative at the telephone number below.

Respectfully submitted,

TUROCY & WATSON, LLP

/Brian Steed/

Brian Steed
Reg. No. 64,095

TUROCY & WATSON, LLP
127 Public Square
57th Floor, Key Tower
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731